1. Components they take the state and props and they output the UI.
   1. We can directly use the components in the html state.
2. Reactive Changes:
   1. The react will automatically handle all the DOM changes and then update the browser.
3. Virtual Views in Memory
   1. We will be focusing on the tree reconciliation.
   2. This is also known as the virtual dom.

There are two types of components

1. Functional Components
2. Class Components

The components is going to take the props & state then provide output as the DOM.

The props are immutable and the state can be changed inside the component.

The html which is present in the JS file is then converted into the jsx

React.createElement(“div”,”container”,”hello world”)

The conversion from html to jsx is done by the help of the babel compiler

The components should always be capitalized.

The Props are in the form of the key value pair.

Props can also hold function as well as data.

We can also pass parameters to the function but we cannot directly do that, It should be done in a function.

The const keyword will make the variable constant but if the content in the variable is an array or object the content in it can be updated.

Array Destructing will helps us in decreasing the code from

PI = Math.PI

E = Math.E

Sqrt = Math.Sqrt

The above code is similar to

const [PI,E,SQRT]= Math

The above code will also do the same thing

In the array destructing, the we can use this “…” to store all the remaining elements into the array.

Const[first, …restOfItems]= [10,20,30,40]

The below is another example of the array destructing

Const data={

Temp1:1,

Temp2: 2,

First:Ashfaq,

Second:Nisar

}

Const[temp1, temp2, …Person] = data;//The remaining properties will create a new object called as person.

Const newArray= […restOfItems]

We can provide styles directly in the html tags but we have to use the two margins inside the style component {{}}

The javascript styling helps us in writing the conditional styling.

The statement this.props will capture all the props.

The event object is present in the OnSubmit and we can use the parameter for the function as the event wrapper.

Instead of using the id option for the element, we can use the react ref element which will help us in the getting the input value.

The Ref will help us get the current.value in the variable

Instead of ref we can also use the state object to read the variable, to get the value we use event.target.value;

We should mostly use the map filter reduce instead of for/while

If you know there is data in the UI and the value change over time then we should definitely try to change them into the state component.

Always create components with two names in the component.

We should always try to minimize the state and use the state to handle other things as well.

The computation should not be done at the using the components. It should be done in an variable and then used if it is true or not.

First, we should have any sort of state and then we should have any sort of hooks and then we can make some computations in the program.

We also have sideEffects in react rather than just state hook.

As the useEffect will rendered whenever there is an change in the UI, the renderedUI will create its own timer, which will lead to issues in the program. So, we have to clean the program whenever we don’t use it .

We should always clean up the effect after creating it.

It is better to have one variable instead of two variables

We should reset all the states, if we have to use the component again.

Instead of resetting all the states, we also unmount the component and remount it

We can do the above following, by the help of the key component. We just need to change the value in the key to reset the previous component and create a new component

We can split all the responsibilities within the application and create different components to handle the different things in the application.

We cannot call hooks inside the loop or conditions

We have to put everything inside the hook for handling everything.

For destructing, we have to use the flower brackets, instead of the square brackets